```
CREATE CONSTRAINT EmailIsUnique on (p:Person) ASSERT p.email IS UNIQUE

CREATE CONSTRAINT UsernameIsUnique on (p:Person) ASSERT p.user IS
UNIQUE

CREATE CONSTRAINT SongIdIsUnique ON (s:Song) ASSERT s.id IS UNIQUE


LOAD CSV WITH HEADERS FROM 'file:///cancionesDef.csv' AS row

MERGE (s:Song {id: toInteger(row.id)})

ON CREATE SET

      s.title = row.title,

      s.artist = row.artist,

      s.bpm = toFloat(row.bpm),

      s.energy = toFloat(row.energy),

      s.idSpotify = row.idSpotify,

      s.genre = row.genre,

      s.preview = row.preview,

      s.cover = row.cover,

      s.date = row.date


LOAD CSV WITH HEADERS FROM 'file:///relacionesDef.csv' AS row

MATCH (s:Song {id: toInteger(row.idA)}) MATCH

(t:Song {id: toInteger(row.idB)})

MERGE (s)-[:RELATED]-(t)


CALL gds.graph.create.cypher(

      'grafoCanciones',

     'MATCH (s:Song) RETURN id(s) as id',

     'MATCH (s:Song)-[:RELATED]-(t:Song) RETURN id(s) AS source, id(t)
AS target')

YIELD graphName AS graph, nodeQuery, nodeCount AS nodes,
relationshipQuery, relationshipCount AS rels
```